Online Multi-keywords Map-search Algorithm Based on Sub-region Method

Zhi Yu[1,a], Zhonghan Sun[1,b], Yinlong Zhu[1,c] and Can Wang[1,d]

[1]Zhejiang Provincial Key Laboratory of Service Robot, College of Computer Science, Zhejiang University, Hangzhou 310027, China

[a]yuzhirenzhe@zju.edu.cn, [b]zhsun@zju.edu.cn, [c]819668276@qq.com, [d]wcan@zju.edu.cn

Keywords: Map-search, Multi-keywords, Sub-region method, Online Search

Abstract. Nowadays, the map-search technology is frequently used by people. However, present map-search service can hardlysatisfy people's growing demands. Providing multiplex service support in search service, such as intuitive search results and multi-keywords search, has become a research focus. This paperproposes a novel online multi-keywords map-search algorithm based on sub-region method. This algorithm makes up a graph of separate nodes and transfers the weight of keyword totheir adjacent nodes. When searching for sub-graph of a single keyword, it finds the node with the max weight then tries to accept or rejectits neighbors step by step. Duringthe search ofsub-graphs of multi-keywords, we must find the node with the max harmonic mean weight and union all the sub-graphs of single keyword it belongs to. This algorithm, based on sub-region method, can search multi-keywords in practical map-search experiments, and return several best resultsat the same time.

Introduction

The search engine plays an important role in internet era. As one of the outstanding branches, the map-search technology,people can use it to findrestaurants, hotels,entertainments, schools, shops and so on.After processing, the search engine will mark severalplaces on the map that meet the query.[1-3].However, what if people want to find restaurants and entertainmentsat the same time?Traditional map-search enginemay not find a place contains these both services for us. What we get are just a few marks on the map. Present map-search service can hardly satisfy people's growing demands. To provide multiplex service support in search service, such as multi-keywords search, has become a research focus.

In this paper, we proposed a novel online multi-keywords map-search algorithm which is based on sub-region method. Instead of presenting search result as a few points in the map, we provide some regions contain multi-key words demands. First, we make up a graph of separate nodes and transfers keyword weight to each other. While searching for sub-graph of a single keyword, we have to find the node which has the max weight then try to accept or abandon its neighbors step by step. While searching for sub-graph of multi-keywords, we have to find the node which has the max harmonic mean weight at first, then union all the sub-graphs of single keyword it is in. The algorithm can search multi-keywords at the same time in practical map-search experiment, and return several best results which are based on sub-region method.

Related Works

Map-searchis a keyword search-based technology on the relational database[4,5]. Keyword search method contains two categories: relationship-basedsearch and tuple-based search.

Relationship-based search can be easilyimplementedby SQL statements. As for tuple-based keyword search, it should construct a spanning tree. BANKS-I[6]proposedthe Reverse Search Algorithmthat processesDijkstraalgorithm on the keywords tuples to find the Shortest Path's shared nodes in the tree as a tree's root.Then it searches the nodes containing keywords put on the root as search results.Since the number of executions of the Dijkstraalgorithm is proportional to the number of tuples that contain the keywords, the query results are in a lower efficiency.BANKS-II[7] provided a Bidirectional expansion algorithm thatimproves the BANKS-I algorithm,but it will cause lower quality problem because it loses some minimum spanning paths in computation stage.DPBF[8]chose a node containing keywords as the starting point , alternately performed its definition grow and merge functions, until all the nodes are packages that contain the keywordin the same group containing one tuple connection tree.

Online Multi-keywords Map-search Algorithm Based on Sub-region Method

Consider a set of n nodesn_i, with m related keyword k_j, where i ∈{1,...,n} and j ∈{1,...,m}.We can get two matrices, N=[n_{ij}] $\in R^{n\times m}$,G=[g_{ij}] $\in R^{n\times m}$, wheren_{ij}∈ {0,1}such that n_{ij}=1 represents node $n_i$has the keyword k_j ,otherwhilen_{ij}=0,g_{ij}∈ {0,1} such that g_{ij}= 1 represents node n_i is connected by one edge otherwhileg_{ij}=0.Now we get a undirected graph with every edge weight is 1,but this can’t realize the goal of map-search return a region. For that purpose, we can introduce the Label propagation algorithm[9]into map-search.

Label propagation algorithmcan propagate the weight of nodes that containing the keyword to the nearest node without any keywords. The weight of nodes can diffuse in the constructed graph of all nodes and weight of all nodeswill become stable at last. Here we set variable λ as diffusion rate, variable t as propagation times. Each time when propagation happens, nodes have keyword will contribute 1-λ to nearest nodes and only leave λ.After t times, the weight of keyword become balance in the constructed graph of all node. Until now, we get a new weight matrix N of keywords.

First, let’s focus on single keyword map-search. After label propagation, map-search will find a set of nodes containing keyword named R.What’s more? Bigger weight represents more original nodes containing keyword nearby.Following this idea, we can find the region of goal by sort the weight of keyword $k_i$and decide whether one node should belong to R one by one using thresholdδ . Here we can also thinkset R as sub-region. Details can be found in the algorithm1.

Algorithm 1 single keyword map-search

Input: keywords weight matrix G,propagation variableλ threshold δ and query keyword k;
Output: a set R of nodes containing keyword;
Initialization: R = null,

1. Label propagation algorithmwith λ
2. Loop:
3. Find the largest weight node of keyword k, push into R
4. Loop:
5. Search the near neighborhood if weight >δ , push into R
6. End
7. Del the largest node
8. End
9. Return R

Now, let’s move to multi-keywords map-search problem. Once we can handle the single keyword map-search, multi-keywords map-search just become an intersection of multi-keywords sub-regionR_i, where i represents the i-th keyword.Fellow this idea, we should find a way to save, load and intersect the sub-region of every keyword. The lazy way to save the result is save the index of every keyword of map-search. But if you want to accelerate the efficiency of algorithm and the least computation, we recommend saving data with the tree structure or sorted or descending order. Later you will know why structured data leads good performance.

One more thing, multi-keywords map-search can get resulted region by intersecting the area of single keyword. But how can computation the joint weights of several keywords?Obviously, taking theGeometric mean of all keywords isn’t a good way. Geometric mean will lead bad result if one special keyword has significant weight and others are very small. Here we use theHarmonic

Mean[10]. Harmonic mean can avoid this imbalance weight problem. Once we get the harmonic mean of all keywords, finding the largest node n_l.Next search all sub-regionsR_i which containing this noden_l, intersect the sub-region containing the node n_j. Finally, we get the intersection sub-region as result. Details can be found in the algorithm2.

Algorithm 2 multi-keywords map-search

Input: keywords weight matrix G, propagation variable λ threshold δ and query keyword k;
Output: a set R of nodes containing multi-keywords;
Initialization: R = null,
1. Loop
2. R_i := single keyword map-search;
3. Save R_i with weight
4. End
5. CalculateGeometric mean of all nodes;
6. Find the largest Geometric meannode n_l;
7. Loop
8. Load and search R_i on n_l,
9. If $R_i$has n_l, push $R_i$into R
10. End
11. Return R

In algorithm 2, we save and load $R_i$frequently and R_i will contain significant nodes inside.Structured data leads good performance as mentioned early.

Experimental Results

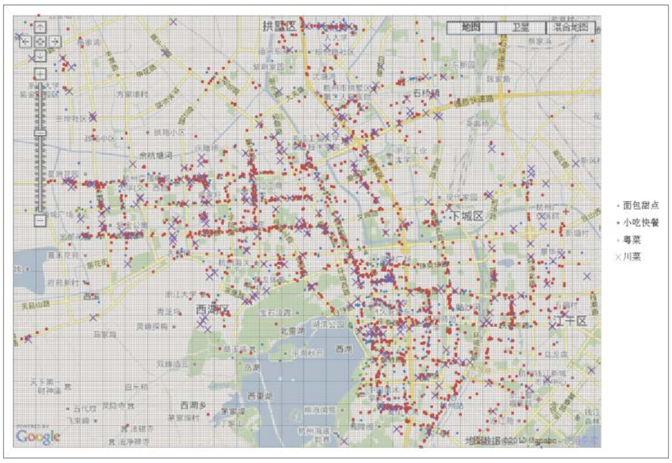

Figure 1 POI example

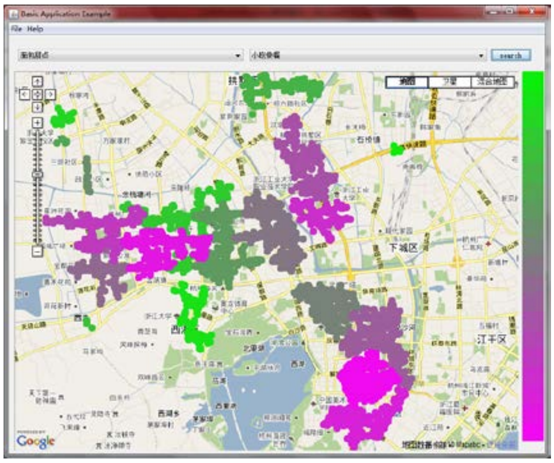

Figure 2Search result example

We use the Google map API to build anonline-systemimplemented in Java.The online-system can process the single keyword search and multi-keywords search.As the figure demonstrates, Fig.1 shows the result of single keyword searchwith single node marker. The results are marked by crosses and dotswhiletheycan't meet users' multi-demands.Fig.2 showsmulti-keywords search results with the colored region. Contrasted with Fig.1, we can easily find the area that meet users' multi-demands and we use the color map show top 4 recommended areas. Warmer color means higher recommend region.

Conclusion

This paperproposes a novel online multi-keywords map-search algorithm which is based on sub-region method. First, we make up graph of separate nodes and transfers keyword weight to each other. While searching for sub-graph of a single keyword, we have to find the node which has the max weight then try to accept or abandon its neighbors step by step. While searching for sub-graph of multi-keywords, we have to find the node which has the max harmonic mean weight at first, then union all the sub-graphs of single keyword it is in. Thisalgorithm can search multi-keywords at the same time. The online-system based onjava and Google Map API can hold single keyword withseveral isolated points results and multi-keywords search with regions results.

Future work

Our algorithm just takes several keywords users provided as equal weight. What if multi-keywords weight in users' demand isdifferent?So we are attempting to improve the online-system to meet users' demands by providing them with regions and several single isolated points at the same time.

Acknowledgement

This work is supported by National Key Technology R&D Program (Grant No. 2012BAI34B01)

References

[1] Hristidis V, Papakonstantinou Y. Discover: Keyword search in relational databases[C]//Proceedings of the 28th international conference on Very Large Data Bases. VLDB Endowment, 2002: 670-681.

[2] Golenberg K, Kimelfeld B, Sagiv Y. Keyword proximity search in complex data graphs[C]//Proceedings of the 2008 ACM SIGMOD international conference on Management of data. ACM, 2008: 927-940.

[3] Lawler E L. A procedure for computing the k best solutions to discrete optimization problems and its application to the shortest path problem[J]. Management Science, 1972, 18(7): 401-405.

[4] HristidisV, Papakonstantinou Y. Discover: Keyword search in relational databases[C]//Proceedings of the 28th international conference on Very Large Data Bases. VLDB Endowment, 2002: 670-681.

[5] Hristidis V, Gravano L, Papakonstantinou Y. Efficient IR-style keyword search over relational databases[C]//Proceedings of the 29th international conference on Very large data bases-Volume 29. VLDB Endowment, 2003: 850-861.

[6] Bhalotia G, Hulgeri A, Nakhe C, et al. Keyword searching and browsing in databases using BANKS[C]//Data Engineering, 2002. Proceedings. 18th International Conference on. IEEE, 2002: 431-440.

[7] Kacholia V, Pandit S, Chakrabarti S, et al. Bidirectional expansion for keyword search on graph databases[C]//Proceedings of the 31st international conference on Very large data bases. VLDB Endowment, 2005: 505-516.

[8] Ding B, Yu J X, Wang S, et al. Finding top-k min-cost connected trees in databases[C]//Data Engineering, 2007. ICDE 2007. IEEE 23rd International Conference on. IEEE, 2007: 836-845.

[9] Wu M, Adviser-Jin R. Label propagation for classification and ranking[M]. Michigan State University, 2007.

[10] Dixon K R, Chapman J A. Harmonic mean measure of animal activity areas[J]. Ecology, 1980: 1040-1044.